

SIEMENS

SIMATIC HMI

WinCC flexible 2005 Funciones de sistema y Runtime Scripting

Impresión de la Ayuda en pantalla

Conceptos básicos	1
Utilizar listas de funciones	2
Elementos y configuración básica	3
Crear scripts	4
Depurar	5
Comportamiento al ejecutar funciones en runtime	6
Ejemplos	7

Impresión de la Ayuda en pantalla

Consignas de seguridad

Este manual contiene las informaciones necesarias para la seguridad personal así como para la prevención de daños materiales. Las informaciones para su seguridad personal están resaltadas con un triángulo de advertencia; las informaciones para evitar únicamente daños materiales no llevan dicho triángulo. De acuerdo al grado de peligro las consignas se representan, de mayor a menor peligro, como sigue.



Peligro

Significa que, si no se adoptan las medidas preventivas adecuadas **se producirá** la muerte, o bien lesiones corporales graves.



Advertencia

Significa que, si no se adoptan las medidas preventivas adecuadas **puede producirse** la muerte o bien lesiones corporales graves.



Precaución

con triángulo de advertencia significa que si no se adoptan las medidas preventivas adecuadas, pueden producirse lesiones corporales.

Precaución

sin triángulo de advertencia significa que si no se adoptan las medidas preventivas adecuadas, pueden producirse daños materiales.

Atención

significa que puede producirse un resultado o estado no deseado si no se respeta la consigna de seguridad correspondiente.

Si se dan varios niveles de peligro se usa siempre la consigna de seguridad más estricta en cada caso. Si en una consigna de seguridad con triángulo de advertencia se alarma de posibles daños personales, la misma consigna puede contener también una advertencia sobre posibles daños materiales.

Personal cualificado

El equipo/sistema correspondiente sólo deberá instalarse y operarse respetando lo especificado en este documento. Sólo está autorizado a intervenir en este equipo el **personal cualificado**. En el sentido del manual se trata de personas que disponen de los conocimientos técnicos necesarios para poner en funcionamiento, conectar a tierra y marcar los aparatos, sistemas y circuitos de acuerdo con las normas estándar de seguridad.

Uso conforme

Considere lo siguiente:



Advertencia

El equipo o los componentes del sistema sólo se podrán utilizar para los casos de aplicación previstos en el catálogo y en la descripción técnica, y sólo asociado a los equipos y componentes de Siemens y de tercera que han sido recomendados y homologados por Siemens. El funcionamiento correcto y seguro del producto presupone un transporte, un almacenamiento, una instalación y un montaje conforme a las prácticas de la buena ingeniería, así como un manejo y un mantenimiento rigurosos.

Marcas registradas

Todos los nombres marcados con ® son marcas registradas de Siemens AG. Los restantes nombres y designaciones contenidos en el presente documento pueden ser marcas registradas cuya utilización por terceros para sus propios fines puede violar los derechos de sus titulares.

Copyright Siemens AG 2004-2005. All rights reserved.

La divulgación y reproducción de este documento, así como el uso y la comunicación de su contenido, no están autorizados, a no ser que se obtenga el consentimiento expreso para ello. Los infractores quedan obligados a la indemnización por daños y perjuicios. Se reservan todos los derechos, en particular para el caso de concesión de patentes o de modelos de utilidad.

Exención de responsabilidad

Hemos comprobado la concordancia del contenido de esta publicación con el hardware y el software descritos. Sin embargo, como es imposible excluir desviaciones, no podemos hacernos responsable de la plena concordancia. El contenido de esta publicación se revisa periódicamente; si es necesario, las posibles correcciones se incluyen en la siguiente edición.

Siemens AG
Automation and Drives
Postfach 4848, 90327 Nürnberg, Alemania

Siemens AG 2005
Sujeto a cambios sin previo aviso

Índice

1	Conceptos básicos	1-1
1.1	Conceptos básicos de "Runtime Scripting"	1-1
1.2	Funciones del sistema	1-2
1.3	Utilizar funciones del sistema	1-4
1.4	Scripts	1-4
1.5	Utilizar scripts.....	1-6
2	Utilizar listas de funciones	2-1
2.1	Conceptos básicos de las listas de funciones	2-1
2.2	Propiedades de la lista de funciones	2-2
2.3	Configurar la lista de funciones.....	2-3
2.4	Editar la lista de funciones	2-5
3	Elementos y configuración básica	3-1
3.1	Scripts	3-1
3.2	Propiedades del editor de scripts	3-3
3.3	Configuración básica	3-6
4	Crear scripts	4-1
4.1	Crear un script nuevo.....	4-1
4.2	Introducir código en un script.....	4-2
4.3	Acceder a variables	4-4
4.4	Llamar a scripts y a funciones del sistema en scripts.....	4-5
4.5	Acceder a objetos	4-6
4.6	Sincronizar variables y objetos	4-7
4.7	Comprobar scripts.....	4-8
4.8	Guardar scripts	4-9
4.9	Cambiar el nombre de un script.....	4-9
5	Depurar	5-1
5.1	Depurar scripts.....	5-1
5.2	Integrar un depurador	5-2
5.3	Iniciar el depurador	5-6
5.4	Manejar el depurador	5-6
5.5	Salir del depurador.....	5-6

6	Comportamiento al ejecutar funciones en runtime.....	6-1
6.1	Procesar la lista de funciones en runtime	6-1
6.2	Procesar scripts en runtime	6-2
6.3	Transferir y retornar valores.....	6-2
6.4	Modificar propiedades de objetos en runtime con VBS.....	6-3
6.5	Funciones del sistema dependientes del panel de operador	6-3
7	Ejemplos.....	7-1
7.1	Ejemplo: Función para convertir temperaturas de Fahrenheit a Celsius.....	7-1
7.2	Ejemplo: Subrutina para convertir pulgadas a metros.....	7-3
7.3	Ejemplo: Cambiar el modo de operación del panel de operador y visualizar el estado actual .	7-4

Índice alfabético

Conceptos básicos

1.1 Conceptos básicos de "Runtime Scripting"

Introducción

Para las tareas de configuración convencionales, WinCC flexible ofrece funciones de sistema predefinidas que permiten resolver muchas tareas en runtime sin necesidad de disponer de conocimientos de programación.

Para otro tipo de tareas se puede recurrir al Runtime Scripting. Runtime Scripting es una interfaz de programación que permite acceder en runtime a algunos datos del proyecto. La aplicación de Runtime Scripting se dirige a ingenieros programadores con conocimientos de Visual Basic (VB) y Visual Basic Script (VBS).

Utilización de funciones del sistema

Las funciones de sistema le serán de utilidad cuando desee vincular elementos de mando a una función:

- Activar un bit en el control
- Cambiar el valor de una variable
- Iniciar una grabación

Las funciones del sistema se pueden utilizar en listas de funciones y scripts.

Utilización de Runtime Scripting

Runtime Scripting está disponible a partir del OP 270/TP 270 y, por tanto, a partir de WinCC flexible Standard. Para ello se soporta el lenguaje de programación VBScript. Runtime Scripting ofrece gran flexibilidad para la configuración. Con Runtime Scripting es posible crear scripts cuando se necesiten funciones adicionales en runtime, tales como:

- Convertir valores
Utilizando scripts es posible convertir valores entre diferentes unidades de medida (p.ej. temperaturas).

- Automatizar procesos de producción

Los scripts permiten controlar procesos de producción, transfiriendo para ello los datos de producción a un autómeta. Utilizando valores de retorno es posible comprobar el estado y, dado el caso, tomar las medidas necesarias.

Scripts

El código VBScript propio se guarda en un script. El script se puede utilizar en el proyecto de igual manera que una función del sistema. En los scripts es posible acceder a las variables del proyecto, así como al modelo de objetos runtime de WinCC flexible. Además, en los scripts se pueden utilizar todas las funciones VBS estándar. En un script se puede acceder a otros scripts y a otras funciones del sistema.

Ejecutar funciones del sistema y scripts

Las funciones del sistema y los scripts se ejecutan en runtime tras ocurrir un evento configurado (p.ej. clic con el ratón en un botón).

Profundidad de recurrencia

La profundidad de recurrencia de los scripts está limitada por la capacidad de pila del panel de operador. Una cantidad ilimitada de recurrencias provoca un aviso de error del sistema en runtime. Por ello, es necesario limitar el número de recurrencias por script.

Ver también

Conceptos básicos de las listas de funciones (Página 2-1)

Utilizar scripts (Página 1-6)

1.2 Funciones del sistema

Introducción

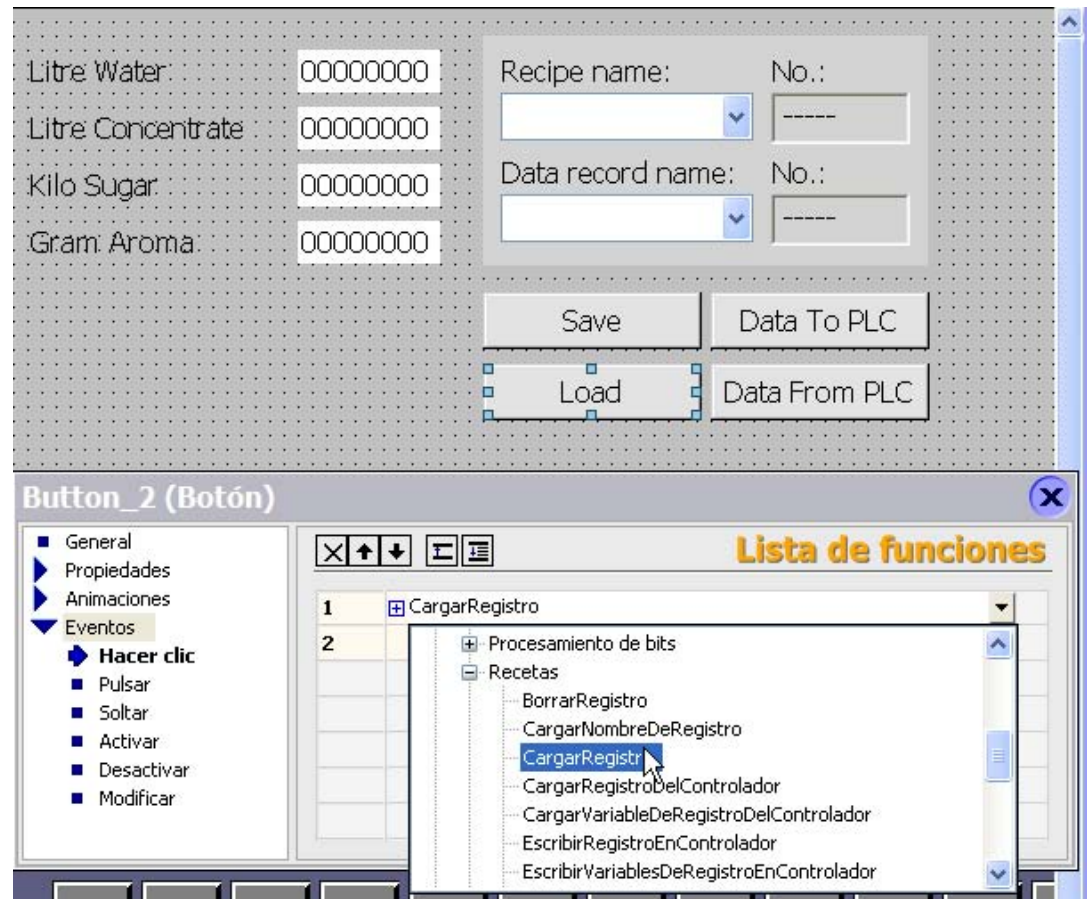
Las funciones del sistema son funciones predefinidas que permiten realizar numerosas tareas en runtime incluso sin tener conocimientos de programación, tales como:

- Cálculos (p.ej. aumentar el valor de una variable en una cantidad fija o variable)
- Funciones de archivar (p.ej. iniciar un fichero de valores de proceso)
- Configuración (p.ej. cambiar de control o activar un bit en el control)
- Avisos (p.ej. tras cambiar de usuario)

Utilización

Las funciones del sistema se pueden utilizar en listas de funciones o en scripts. Puesto que se trata de funciones predefinidas, las funciones del sistema no se pueden modificar.

Cuando configure una lista de funciones, seleccione las funciones del sistema en una lista de selección ordenada por categorías:



Si desea utilizar una función del sistema en un script, selecciónela en la lista. Para acceder a la lista de selección en el script, pulse las teclas <CTRL+barra espaciadora>

Dependencia del idioma

Los nombres de las funciones del sistema dependen del idioma de configuración ajustado. Gracias a ello, el ingeniero programador podrá reconocer las funciones de inmediato.

Excepción: Si desea llamar a una función del sistema en un script, utilice el nombre inglés de la misma. Los nombres ingleses de las funciones del sistema se indican en la referencia de las mismas.

Disponibilidad

En WinCC flexible sólo se pueden configurar funciones soportadas por el panel de operador seleccionado. Por tanto, en una lista de funciones sólo podrá configurar las funciones del sistema que soporte el panel de operador en cuestión. Si desea utilizar un proyecto en varios paneles de operador, las funciones del sistema no soportadas por un panel en particular se identificarán en color.

Ver también

Conceptos básicos de las listas de funciones (Página 2-1)

1.3 Utilizar funciones del sistema

Introducción

En runtime se ejecuta una lista de funciones cuando ocurre el evento configurado. El operador puede lanzar un evento p.ej. pulsando una tecla de función en el panel de operador. El sistema también puede disparar eventos p.ej. cuando un valor de proceso excede un límite inferior.

Posibilidades de aplicación

Las funciones del sistema se pueden configurar en todos los objetos que reaccionen a eventos. Además, se pueden utilizar directamente en listas de funciones y scripts, lo que permite controlar el proceso.

- Lista de funciones

Las funciones del sistema se procesan en las listas de funciones de forma secuencial, es decir, desde la primera hasta la última función. Para evitar tiempos de espera, las funciones del sistema de ejecución prolongada (p.ej. operaciones con archivos) se procesan de manera paralela. Gracias a ello, la función del sistema siguiente se puede comenzar a ejecutar aunque no haya finalizado todavía la función anterior.

Para más información sobre cómo configurar una lista de funciones, consulte el ejemplo "Cambiar el modo de operación del panel de operador y visualizar el estado actual".

- Script

En los scripts es posible utilizar funciones del sistema combinadas con instrucciones y condiciones del código. Gracias a ello, un script se puede ejecutar dependiendo de un determinado estado del sistema. Además, es posible evaluar p.ej. valores de retorno de las funciones del sistema. Dependiendo del valor de retorno, podrá ejecutar entonces p.ej. funciones de comprobación que, a su vez, influyen en el procesamiento del script.

1.4 Scripts

Introducción

Los scripts se programan en código VB-Script. Un script listo se puede utilizar en el proyecto de igual manera que una función del sistema. Al crear un script es preciso determinar su tipo y definir los parámetros de transferencia. Los scripts del tipo "Función" tienen un valor de salida. Los scripts del tipo "Sub" se denominan "Procedures" y no tienen valor de retorno.

Propiedades de los scripts

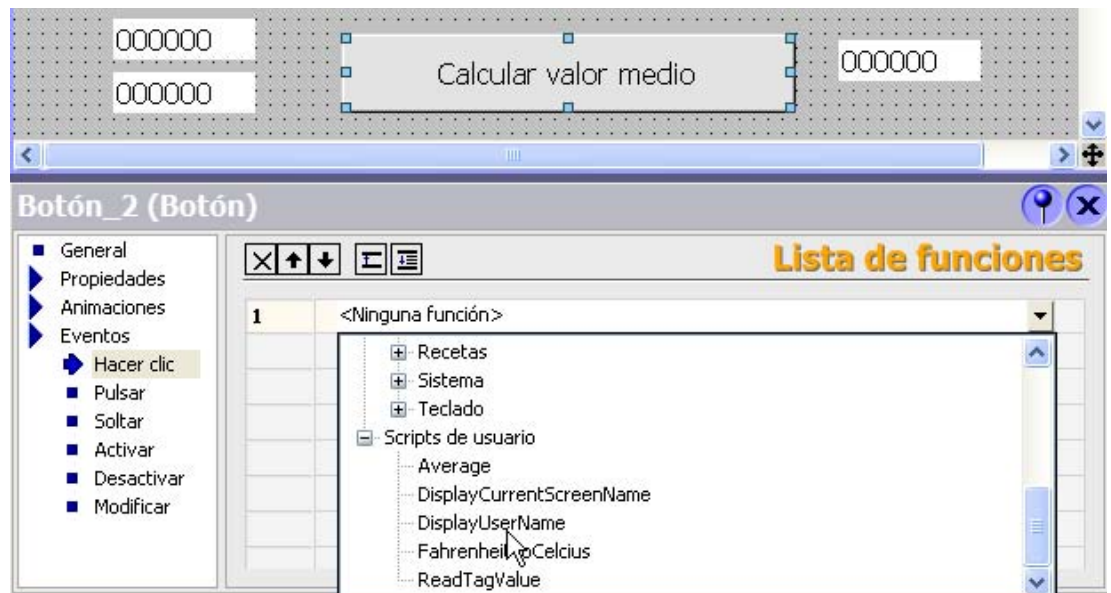
Desde un script se pueden ejecutar otros scripts y funciones del sistema. A través del modelo de objetos runtime es posible acceder a los objetos runtime de WinCC flexible. Al utilizar funciones del sistema en un script, utilice los nombres ingleses de las mismas. En los scripts se puede utilizar toda la funcionalidad idiomática de Microsoft VBScript, exceptuando las funciones y métodos de interacción con el usuario, tales como "MsgBox".

Si utiliza en un script funciones del sistema no disponibles en el panel de operador seleccionado, aparecerá un aviso de advertencia. Además, la función del sistema en cuestión se subrayará en el script con una línea ondulada en color azul.

Organizar scripts

Los scripts se guardan en la base de datos del proyecto. Los scripts disponibles se visualizan en la lista "Scripts" en la ventana del proyecto.

Si desea utilizar un script en una lista de funciones, seleccione el script deseado en la lista de selección "Scripts".



Ver también

Scripts (Página 3-1)

Crear un script nuevo (Página 4-1)

1.5 Utilizar scripts

Principio

Los scripts ofrecen gran flexibilidad gracias a que utilizan elementos de control de un lenguaje de programación.

Los scripts permiten crear en runtime soluciones personalizadas en un proyecto, tales como:

- Configurar la lista de funciones ampliada

Un script se puede utilizar de igual manera que una lista de funciones, llamando allí funciones del sistema y otros scripts.

En un script es posible ejecutar o repetir funciones del sistema y scripts conforme a las condiciones predeterminadas. A continuación, el script se agrega a una lista de funciones.

- Programar nuevas funciones

Los scripts están disponibles en el proyecto entero. Los scripts se pueden utilizar de igual manera que las funciones del sistema. Para estos scripts puede definir parámetros de transferencia y valores de retorno. Los scripts se pueden utilizar p.ej. para convertir valores.

Ver también

Conceptos básicos de "Runtime Scripting" (Página 1-1)

Utilizar listas de funciones

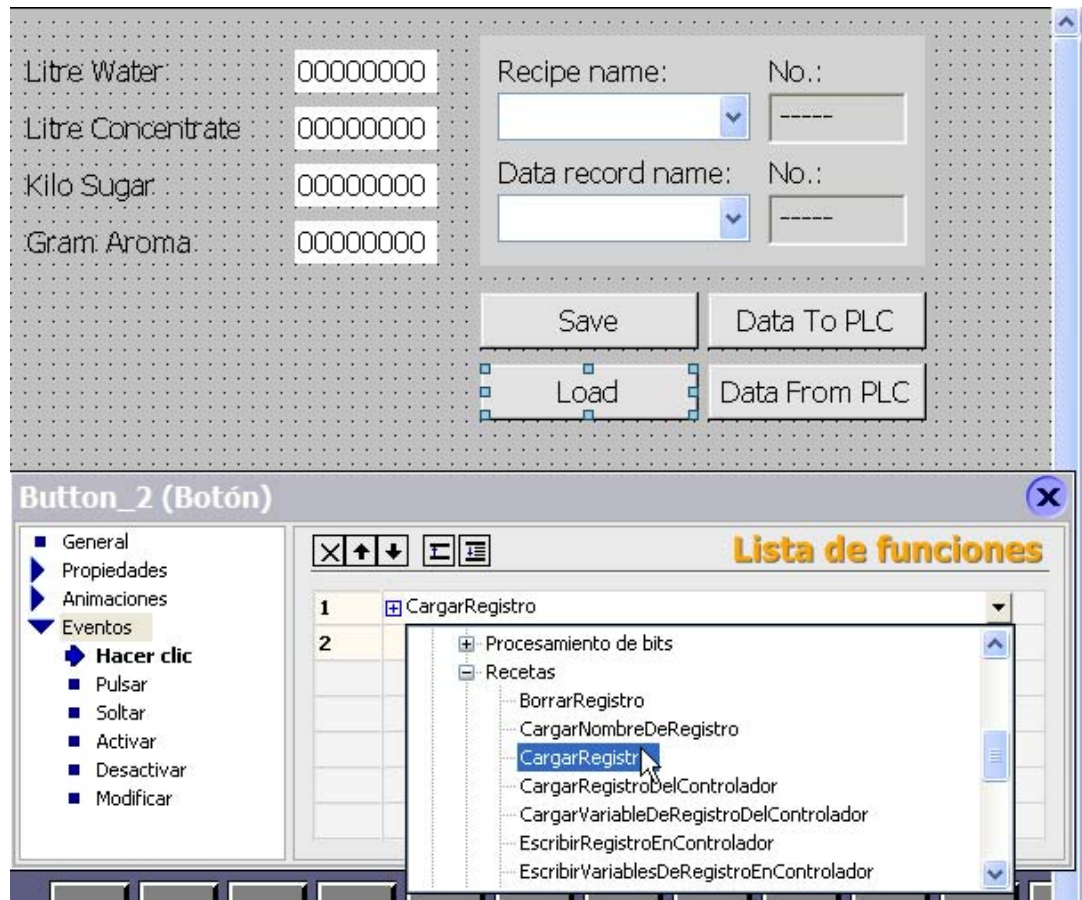
2.1 Conceptos básicos de las listas de funciones

Introducción

Una lista de funciones permite ejecutar varias funciones del sistema y scripts cuando ocurra el evento configurado.

Principio

La lista de funciones se configura en el evento de un objeto (p. ej. un objeto de imagen o una variable). Los eventos disponibles dependen del objeto seleccionado y del panel de operador utilizado.



Los eventos sólo ocurren cuando el proyecto está en runtime. Los eventos pueden ser:

- Cambiar el valor de una variable
- Pulsar un botón
- Activar runtime

En cada evento se puede configurar exactamente una lista de funciones.

Nota

Las funciones del sistema configurables que se pueden seleccionar en una lista de funciones dependen del panel de operador utilizado.

Ver también

Funciones del sistema (Página 1-2)

Conceptos básicos de "Runtime Scripting" (Página 1-1)

2.2 Propiedades de la lista de funciones

Dependencia del panel de operador

Un mismo proyecto se puede utilizar en diversos paneles de operador. Si cambia el panel de operador en el proyecto, todas las funciones del sistema y scripts no soportados por el panel en cuestión se destacarán en color amarillo. Las funciones del sistema no soportadas no se ejecutarán en runtime.

Información de estado

Durante la configuración, los datos del proyecto se comprueban en segundo plano. La información de estado indica en cada lista de funciones el estado de las funciones del sistema y de los scripts contenidos en ella.

Las informaciones de estado significan lo siguiente:

- Naranja: La lista de funciones no se ejecuta en runtime, puesto que por lo menos una función del sistema o un script no se ha parametrizado por completo.
- Amarillo: La lista de funciones se ejecuta en runtime. No obstante, contiene por lo menos una función del sistema o un script no soportado por el panel de operador seleccionado (p.ej. debido a un cambio del tipo de equipo).

Procesar funciones del sistema y scripts

Las funciones del sistema y los scripts contenidos en una lista de funciones se procesan en runtime de forma secuencial, es decir, de arriba a abajo. Para evitar tiempos de espera, las funciones del sistema de ejecución prolongada (p.ej. operaciones con archivos) se procesan de manera paralela. Gracias a ello, la función del sistema siguiente se puede comenzar a ejecutar aunque no haya finalizado todavía la función anterior.

Para programar procesos no secuenciales y condicionados, utilice un script con bucles, instrucciones condicionadas y condiciones de interrupción.

2.3 Configurar la lista de funciones

Introducción

Para configurar una lista de funciones, seleccione las funciones del sistema y los scripts en la lista de selección correspondiente. Las funciones del sistema están ordenadas por categorías en la lista de selección. Los scripts se encuentran en la lista de selección bajo la entrada "Scripts".

Nota

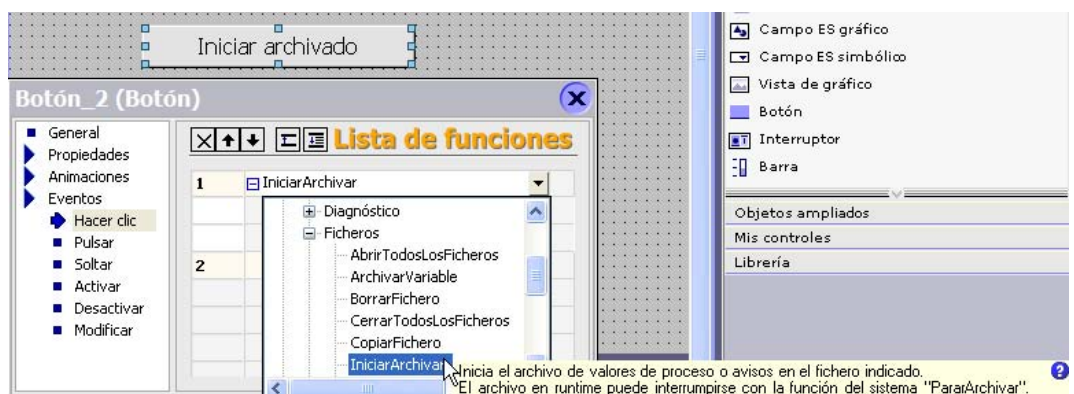
Para agregar funciones del sistema o scripts al "Asistente de scripts", proceda de igual manera que al configurar una lista de funciones. Abra para ello el asistente de scripts y seleccione las funciones del sistema y los scripts, o bien insértelos con el método de copiar y pegar.

Requisito

El objeto deberá tener por lo menos un evento configurable.

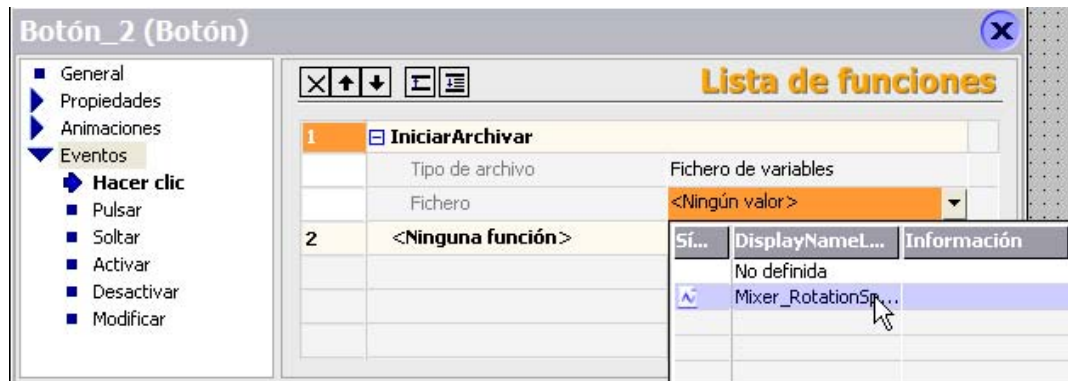
Procedimiento

1. En WinCC flexible, abra el editor que contiene el objeto.
2. Seleccione el objeto con el ratón.
3. En la ventana de propiedades del grupo "Eventos", haga clic en el evento que desea configurar en la lista de funciones.
4. En la lista de selección de la ventana de propiedades, seleccione la entrada "<Ninguna función>" con el ratón.
5. En la lista de selección, escoja la función del sistema o el script deseado. También puede introducir directamente el nombre de la función del sistema o del script.



La función del sistema o el script se introducirá en la lista de funciones.

- Si se ha parametrizado la función del sistema o el script, seleccione los valores correspondientes para los parámetros.



- Si desea agregar más funciones del sistema o scripts a la lista de funciones, repita los pasos 4 a 7.

Resultado

La lista de funciones deberá estar configurada. El estado de la lista de funciones se visualizará en la ventana de propiedades junto al evento configurado. Cuando el evento configurado ocurra en runtime, la lista de funciones se procesará de arriba a abajo.

Ver también

- Editar la lista de funciones (Página 2-5)
- Propiedades de la lista de funciones (Página 2-2)

2.4 Editar la lista de funciones

Introducción

La lista de funciones se puede editar como se indica a continuación:

- Modificar el orden de procesamiento de las funciones del sistema o de las funciones
- Eliminar una función del sistema o un script

En el apartado "Configurar la lista de funciones" se describe cómo agregar una función del sistema o un script a la lista de funciones.

Requisito

La lista de funciones deberá estar configurada.

Procedimiento

1. En WinCC flexible, abra el editor que contiene el objeto.
2. Seleccione el objeto con el ratón.
3. En la ventana de propiedades del grupo "Eventos", haga clic en el evento cuya la lista de funciones desea editar.
4. Para modificar el orden de procesamiento de la lista de funciones, seleccione con el ratón la función del sistema o el script deseado en la lista de selección.
5. A continuación, en la ventana de propiedades haga clic con el ratón en la flecha de sentido correspondiente hasta que la función del sistema o el script aparezca en la posición deseada.



Figura 2-1 Desplazar una función del sistema o un script

- Para eliminar una función del sistema o un script de la lista de funciones, seleccione con el ratón la función del sistema o el script en cuestión. A continuación, en la ventana de propiedades haga clic en el botón "Eliminar".



Figura 2-2 Eliminar una función del sistema o un script de una lista de funciones

Procedimiento alternativo

En la lista de funciones también es posible seleccionar varias funciones del sistema y scripts si desea desplazarlas simultáneamente. Pulse y mantenga oprimida la tecla <CTRL> y seleccione con el ratón las funciones del sistema y los scripts que desea desplazar.

Para desplazarlos puede utilizar asimismo el método de arrastrar y soltar.

Ver también

Configurar la lista de funciones (Página 2-3)

3

Elementos y configuración básica

3.1 Scripts

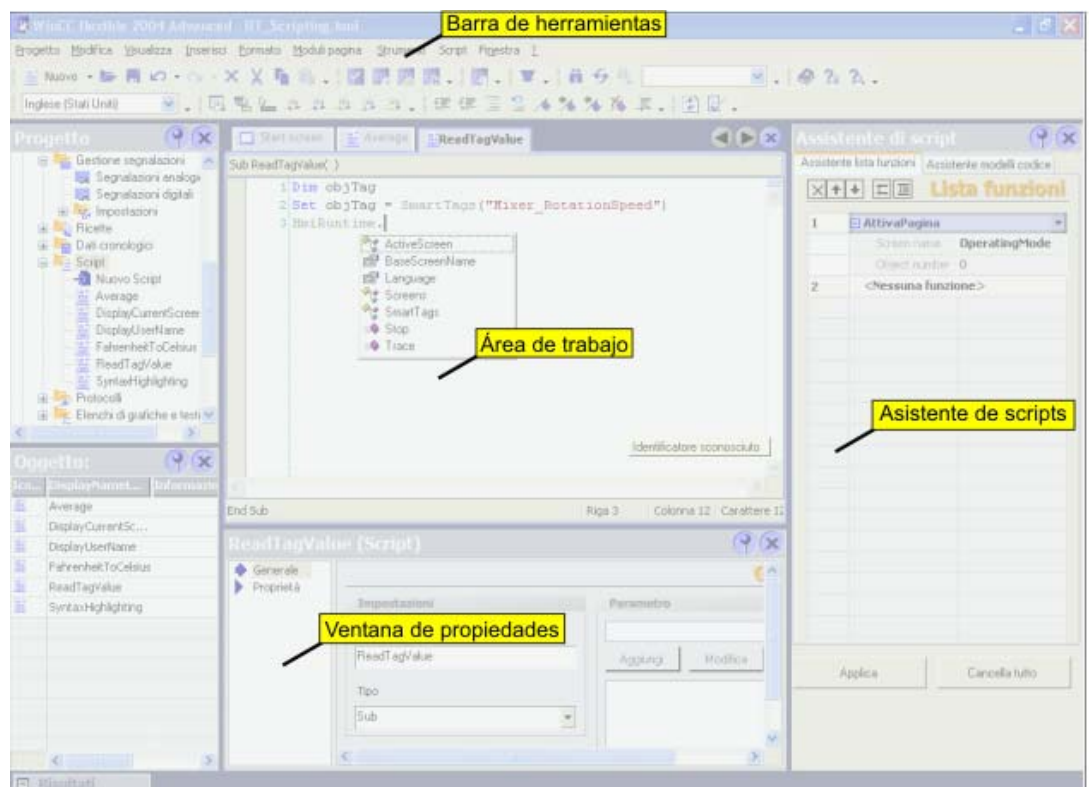
Introducción

Los scripts se editan y se crean en el editor de scripts.

Abrir

El editor de scripts se abre automáticamente al crear un nuevo script, o bien al abrir un script ya existente.

Construcción



Barra de menús

En la barra de menús se encuentran todos los comandos para el manejo de WinCC flexible. Las combinaciones de teclas disponibles aparecen junto a los comandos de menú.

Barra de herramientas "Scripts"

En la barra de herramientas "Scripts" encontrará los comandos para la sincronizar objetos y variables, así como para comprobar la sintaxis de los scripts.

Barra de herramientas "Edición avanzada"

En la barra de herramientas "Edición avanzada" encontrará los comandos para trabajar con marcadores, insertar o extraer códigos, agregar comentarios sobre el código y saltar a una línea de código determinada.

Barra de herramientas "IntelliSense"

La barra de herramientas "IntelliSense" contiene los comandos para visualizar listas de selección, p.ej. todos los objetos del modelo de objeto, las funciones del sistema disponibles o las constantes VBS.

Área de trabajo

Los scripts se crean y se editan en el área de trabajo. Para facilitar la creación de scripts se utilizan la marcación de sintaxis e IntelliSense.

Ventana de propiedades

Los scripts se configuran en la ventana de propiedades. Es posible determinar si el script se referirá a un procedimiento o a una función. Además, puede declarar parámetros para el script.

"Asistente de scripts"

En el "Asistente de scripts" es posible almacenar funciones del sistema y scripts como en una lista de funciones con parámetros asignados. Las funciones del sistema y los scripts almacenados se pueden aplicar en el script activo desde el "Asistente de scripts". De esta forma sólo tendrá que asignar los parámetros una sola vez.

Si ya ha configurado funciones del sistema o scripts en eventos, puede transferirlos al "Asistente de scripts" con el método de copiar y pegar. En dicho asistente sólo podrá almacenar las funciones del sistema permitidas en los scripts. Si intenta transferir con el método de copiar y pegar funciones del sistema que no se puedan utilizar en scripts, dichas funciones se identificarán en color.

Ver también

Crear un script nuevo (Página 4-1)

Propiedades del editor de scripts (Página 3-3)

Configuración básica (Página 3-6)

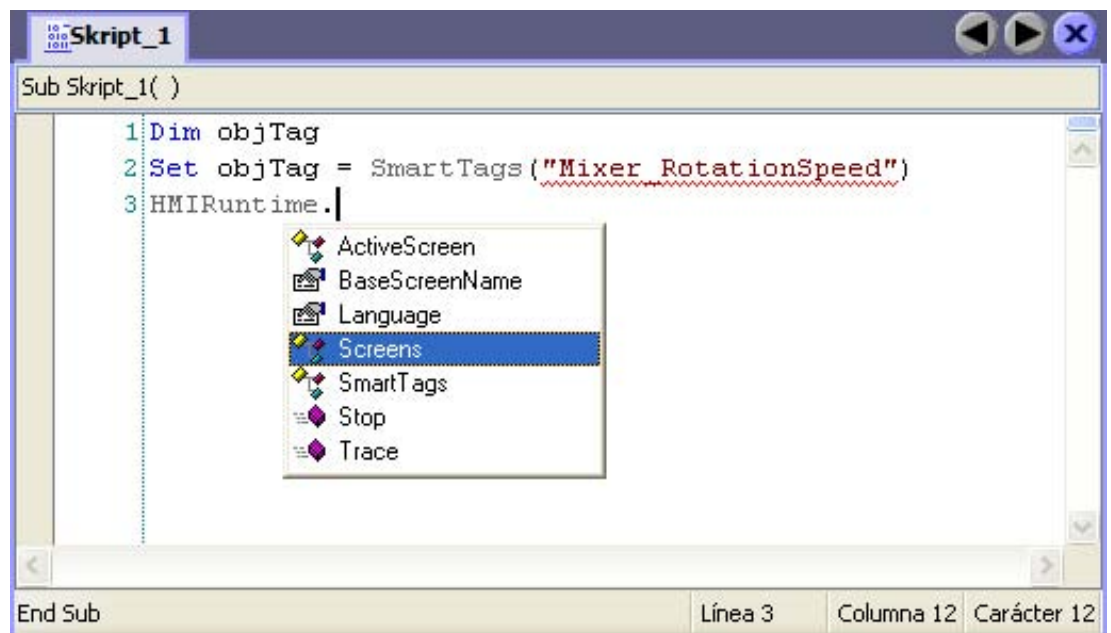
3.2 Propiedades del editor de scripts

Introducción

El editor de scripts le será de utilidad en la programación con funciones como IntelliSense, marcación de sintaxis, etc. Por ejemplo, puede crear referencias a variables con el método de arrastrar y soltar.

IntelliSense

IntelliSense sirve para acceder a objetos, métodos o propiedades del modelo de objetos VBS:



En la lista de selección puede elegir los métodos y propiedades disponibles para el objeto en cuestión.

Marcación de sintaxis

En el editor de scripts, las palabras clave se resaltan en distintos colores. Los objetos reconocidos por el editor de scripts se representan en negrita. Las palabras desconocidas se subrayan con una línea ondulada roja:



La tabla muestra los colores predeterminados para las palabras clave más importantes:

Color	Descripción	Ejemplo
Azul	Palabra clave (VBS)	Dim, If, Then
Gris	Palabra clave (modelo de objeto)	HmiRuntime
Cian	Script	FahrenheitToCelsius
Marrón	Función del sistema	IncreaseValue
Rojo	Variable	Tag_1
Verde	Comentario	'Esto es un comentario

Sincronizar objetos

Las instancias de objetos (p.ej. variables) se sincronizan automáticamente con los datos de configuración al abrir el script. Por ejemplo, si cambia el nombre de una variable en el editor "Variables", esta modificación también tendrá efecto en el script. Si efectúa un cambio estando abierto el script afectado, el objeto cuyo nombre haya modificado aparecerá subrayado con una línea ondulada azul. Entonces podrá llevar a cabo la sincronización de forma manual en el editor de scripts.

Lista de objetos

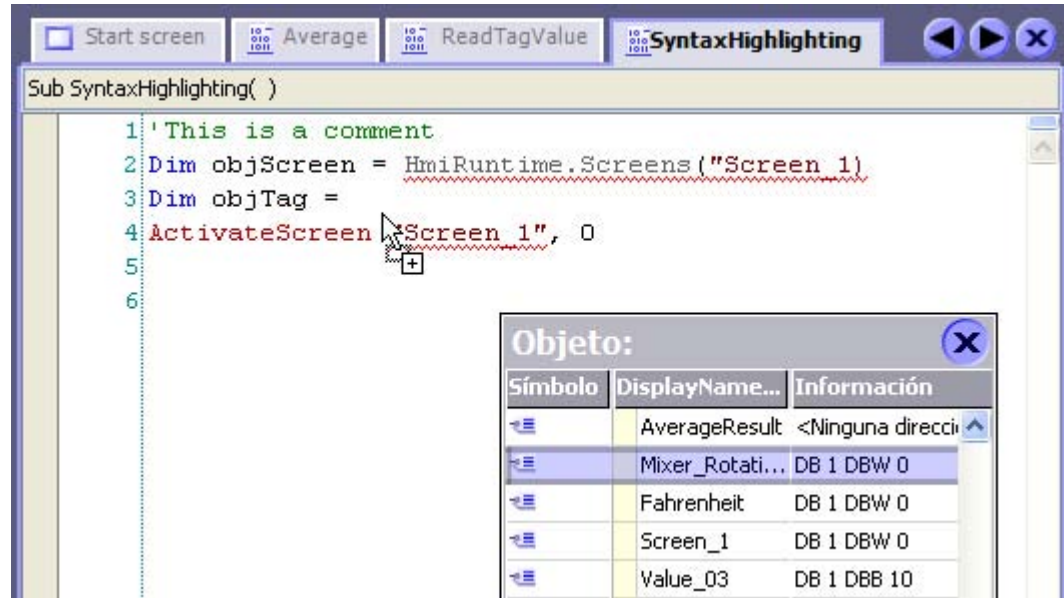
Con la combinación de teclas <Alt+Derecha> puede llamar a la lista de objetos en la que se visualizan todos los objetos disponibles en el contexto determinado. La lista de objetos se puede abrir, por ejemplo, durante la asignación de parámetros o al utilizar listas.

Ejemplo: Desea referenciar una imagen de proceso ya existente a través de la lista de imágenes. En el editor de scripts, introduzca "HmiRuntime.Screens" y, a continuación, abra la lista de objetos pulsando la combinación de teclas <Alt+Derecha>. Allí aparecerán en una lista todas las imágenes de proceso disponibles en el proyecto:

Seleccione la imagen de proceso deseada y confirme su selección pulsando la tecla <Intro>.

Arrastrar y soltar

Si necesita una variable en el script, podrá arrastrarla desde la ventana de objetos hasta el script.

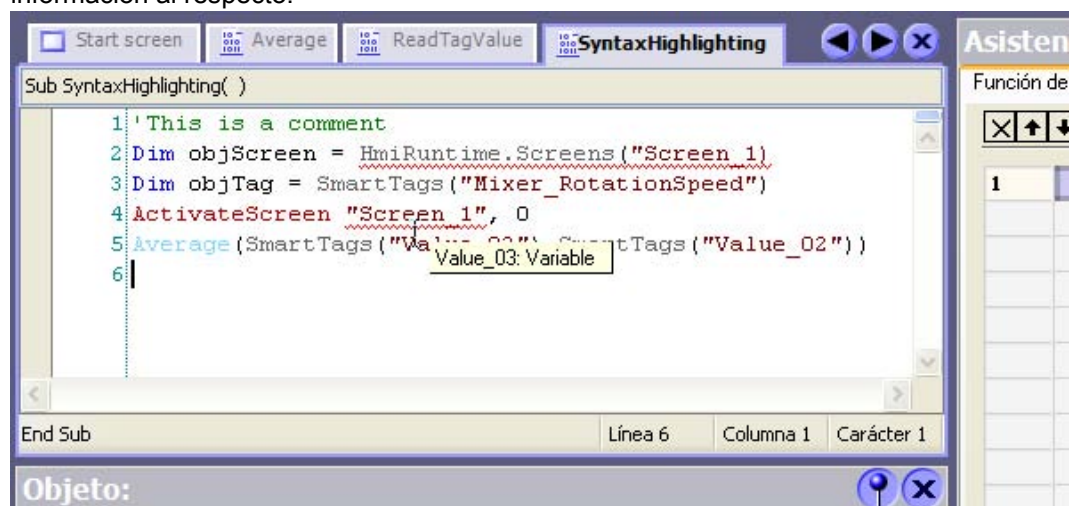


Funciones de ayuda

Durante la programación aparecen automáticamente descripciones breves de los parámetros necesarios para los métodos y funciones del sistema. Además, en el editor de scripts están disponibles las funciones de ayuda indicadas a continuación:

- Información sobre herramientas

Las palabras clave desconocidas o escritas incorrectamente aparecerán subrayadas con una línea ondulada. Al mover el puntero del ratón sobre la palabra aparecerá una información al respecto:



Tratándose de palabras clave conocidas, la información de herramientas mostrará el tipo de palabra clave.

3.3 Configuración básica

- Información sobre parámetros

La información sobre parámetros informa acerca de la sintaxis y los parámetros de una función del sistema o de una función VBS estándar.

- Ayuda contextual

La ayuda contextual (o ayuda sensible al contexto) proporciona información acerca de las funciones del sistema, los elementos de idioma VBScript, los objetos, etc.

Si necesita información sobre un objeto, un método o una propiedad, coloque el puntero del ratón sobre la palabra clave correspondiente y pulse <F1>. Así saltará automáticamente al tema correspondiente de la ayuda en pantalla.

Ver también

Scripts (Página 3-1)

3.3 Configuración básica

Introducción

La configuración básica del editor de scripts se puede personalizar. En el menú "Herramientas", seleccione el comando "Configuración". Modifique la configuración básica en el grupo "Scripts".

Modificar la configuración del editor

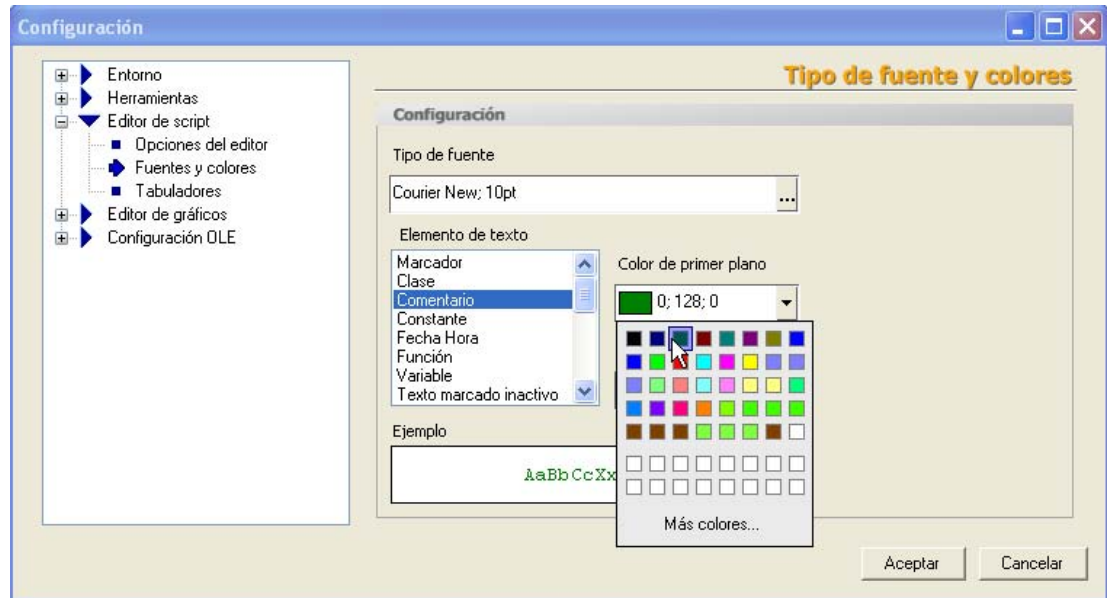
En el script es posible modificar la configuración del editor. Por ejemplo, es posible ocultar la visualización del texto de ayuda en el script.

Para modificar la configuración del editor, en el grupo "Scripts" haga clic en la entrada "Configuración del editor".

Modificar el resalto de sintaxis

Puede cambiar el color predeterminado de todos los elementos de texto. Para ello, en el grupo "Scripts" haga clic en la entrada "Fuentes y colores".

En la lista de selección "Elemento de texto", seleccione el elemento de texto deseado. A continuación, seleccione el color deseado o defina un nuevo color.



Modificar el formato de código

Para modificar el formato de código, proceda de la manera siguiente:

- Para modificar el tipo de letra o su tamaño, en el grupo "Scripts" haga clic en la entrada "Fuentes y colores".
- Para modificar las sangrías o tabulaciones, en el grupo "Scripts" haga clic en la entrada "Sangría".

Ajustes para la transferencia a un panel de operador

Los siguientes ajustes de runtime permiten ahorrar espacio en la memoria tras transferir scripts al panel de operador:

- Transferir informaciones de nombres
- Visualizar comentarios del script

En lugar de nombres de objeto, se transfieren números. Utilice este ajuste únicamente si se han comprobado los scripts y no presentan errores. Si comprueba los scripts en el depurador, el código resultará más claro debido a que se representan los nombres de objetos y los comentarios.

Ver también

Scripts (Página 3-1)

Crear scripts

4.1 Crear un script nuevo

Introducción

Al crear un script nuevo deberá definir los ajustes indicados a continuación:

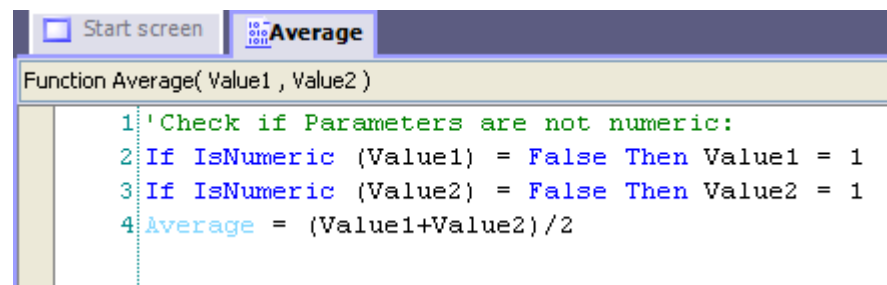
- El nombre con el que se debe llamar al script.
- El tipo de script.
- Los parámetros a transferir al script en runtime.

Procedimiento

1. En la ventana del proyecto, seleccione con el ratón el elemento "Scripts".
2. En el menú contextual de "Scripts", seleccione el comando "Añadir script".
El script se creará en una nueva ficha en el área de trabajo. En la ventana de propiedades se abrirá la máscara para introducir los ajustes de configuración del script.
3. En la ventana de propiedades, introduzca un nombre descriptivo para el script.
4. Seleccione el tipo de script en la ventana de propiedades.
5. Si desea utilizar parámetros en el script, introduzca en la ventana de propiedades los nombres de los parámetros y haga clic en "Añadir".

Resultado

El script se creará. En la barra de título del script se visualizarán el tipo, el nombre y los parámetros del mismo.



```
Function Average( Value1 , Value2 )  
1 'Check if Parameters are not numeric:  
2 If IsNumeric (Value1) = False Then Value1 = 1  
3 If IsNumeric (Value2) = False Then Value2 = 1  
4 Average = (Value1+Value2)/2
```

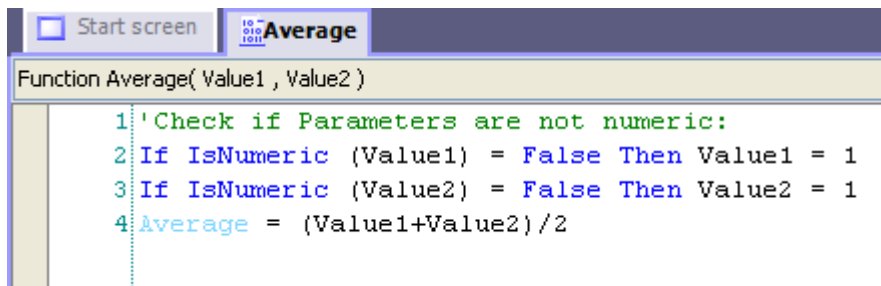
Atención

Los parámetros y el tipo del script sólo se podrán modificar si éste no se ha configurado todavía en la lista de funciones, o si no se ha llamado aún desde otro script.

4.2 Introducir código en un script

Introducción

En el script sólo es preciso introducir el núcleo del código. Las cabeceras del procedimiento y de la función ya se han definido en la declaración en la ventana de propiedades. Por tanto, al final del script tampoco es preciso introducir las instrucciones "End Sub" ni "End Function".



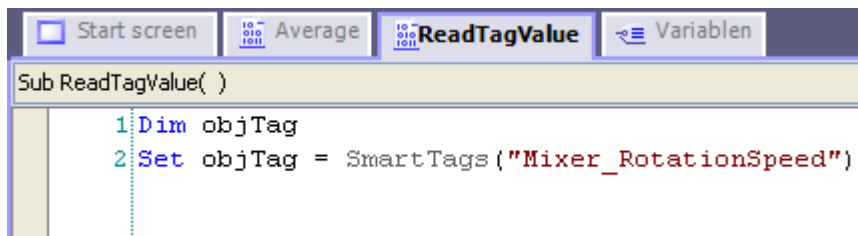
```
Function Average( Value1 , Value2 )  
1 'Check if Parameters are not numeric:  
2 If IsNumeric (Value1) = False Then Value1 = 1  
3 If IsNumeric (Value2) = False Then Value2 = 1  
4 Average = (Value1+Value2)/2
```

Requisito

El script se deberá haber creado.

Utilizar variables del proyecto en el script

Para utilizar una variable del proyecto en un script, desplázela con el método de arrastrar y soltar desde la ventana del objeto hasta la línea de código deseada en el script. La variable del proyecto se insertará utilizando la lista de "SmartTags".



```
Sub ReadTagValue( )  
1 Dim objTag  
2 Set objTag = SmartTags("Mixer_RotationSpeed")
```

Las variables también se pueden insertar en el script a través de la lista de objetos. Para acceder a la lista de objetos, pulse la combinación de teclas <Alt+Derecha>.

Llamar a scripts y funciones del sistema en el script

Para llamar a una función del sistema o a un script, proceda de la manera siguiente:

<Nombre de la función> [Parámetro1], [Parámetro2], [...]

Ejemplo: IncreaseValue FillLevel, 5

En las funciones del sistema y en las funciones estándar VBScript, los parámetros se describen en forma de información de parámetro.

Utilizando la combinación de teclas <CTRL+barra espaciadora> podrá abrir en el script una lista de selección en la que figuran todos los objetos, métodos, funciones del sistema y funciones estándar VBScript que se pueden utilizar en el contexto actual.

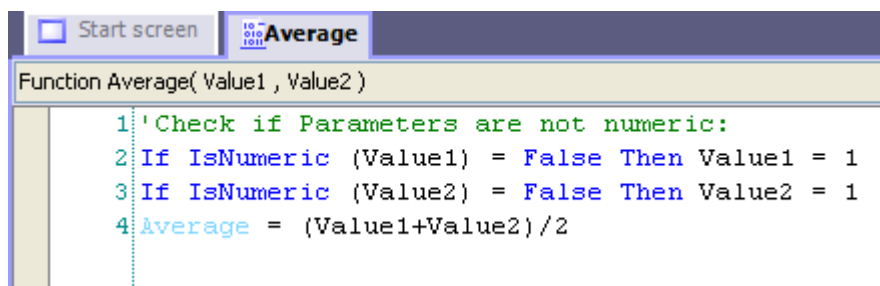
Utilizar las funciones de edición del editor

Utilice los botones de la barra de herramientas "Edición avanzada" para visualizar más claramente el código con sangrías y comentarios. Los marcadores facilitan la orientación incluso si el script contiene muchas líneas de código.

Para más información sobre cada uno de los botones, consulte la ayuda directa correspondiente.

Procedimiento

1. En la ventana del proyecto, haga doble clic en el script en el que desea introducir el código.
El script se abrirá.
2. Introduzca el código.
3. Si el script es del tipo "Function", asigne el valor de retorno al nombre de la función.



```

Function Average( Value1 , Value2 )
1 'Check if Parameters are not numeric:
2 If IsNumeric (Value1) = False Then Value1 = 1
3 If IsNumeric (Value2) = False Then Value2 = 1
4 Average = (Value1+Value2)/2
    
```

Resultado

El script acabado se podrá llamar desde cualquier otro script. Además, se podrá utilizar en listas de funciones de igual manera que una función del sistema.

Ver también

- Llamar a scripts y a funciones del sistema en scripts (Página 4-5)
- Sincronizar variables y objetos (Página 4-7)
- Acceder a objetos (Página 4-6)
- Acceder a variables (Página 4-4)
- Depurar scripts (Página 5-1)
- Procesar la lista de funciones en runtime (Página 6-1)

4.3 Acceder a variables

Introducción

En el script tiene acceso a variables externas e internas que haya creado en el proyecto. En runtime puede leer o modificar el valor de una variable. Además, puede crear variables locales en el script, en calidad de contadores o de memoria intermedia.

Variables del proyecto

Si el nombre de la variable en el proyecto cumple con las convenciones para nombres VBS, podrá utilizar la variable directamente en el script.

```
'VBS_Example_03  
If BeltDriveOilTemperature > 100 Then [instrucción]
```

Si el nombre de la variable en el proyecto no cumple con las convenciones para nombres VBS, deberá referenciar la variable a través de la lista de "SmartTags". En el ejemplo siguiente, el nombre de la variable contiene el carácter "&". Éste no está permitido en las convenciones para nombres VBS.

```
'VBS_Example_04  
Dim objTag  
Set objTag = SmartTags("Test&Trial")
```

Las convenciones para nombres VBS se describen en la ayuda en pantalla del sistema de información VBS.

Variables locales

En el script es posible definir variables locales con la instrucción "Dim". Las variables locales sólo se pueden utilizar dentro del script. Por tanto, no aparecen en el editor "Variables".

En el script, utilice p.ej. una variable local como contador en una instrucción "For".

```
'VBS_Example_05Dim intCountFor intCount = 1 To 10[Instrucción]Next
```

Nota

Cuando requiera el uso de "instrucciones FOR", debe utilizar variables locales. No está permitido utilizar variables de proyecto en las "instrucciones FOR".

4.4 Llamar a scripts y a funciones del sistema en scripts

Principio

En un script es posible llamar funciones del sistema y otros scripts.

Para llamar a una función del sistema o a un script sin valor de retorno ("Sub"), proceda de la manera siguiente:

```
<Nombre de la función> [Parámetro1], [Parámetro2], [...]
```

Para llamar a una función del sistema o a un script con valor de retorno ("Function"), cree una asignación a una expresión:

```
<Expresión> = <Nombre de la función> ([Parámetro1, Parámetro2,  
...[ParámetroN])
```

Si no desea evaluar el valor de retorno, utilice la llamada como si se tratara de una función del sistema o de un script sin valor de retorno.

Particularidades al llamar a funciones del sistema

Las funciones de sistema y los scripts se pueden adoptar del asistente de scripts en el propio script. En el asistente de scripts se muestran las funciones de sistema en el idioma de configuración seleccionado.

Si desea llamar a una función del sistema en un script, deberá utilizar siempre el nombre inglés de la misma.

```
SetValue Tag1, 64
```

Los nombres ingleses de las funciones del sistema se indican en la referencia de las mismas bajo "Sintaxis". El idioma configurado para el proyecto no se tiene en cuenta.

Tenga en cuenta las reglas siguientes al transferir parámetros en las funciones del sistema:

- constantes

Si utiliza una constante como parámetro, el tipo de parámetro deberá corresponder a uno de los tipos de datos indicados a continuación: "Integer", "Double" o "String". Al asignar los parámetros, las constantes disponibles se visualizarán en una lista de selección. Las convenciones VBS habituales son aplicables a las constantes.

- Variables

Independientemente de su notación, las variables se transfieren siempre conforme al principio "Call by Reference". Si la variable a transferir cumple con las convenciones para nombres VPS, podrá transferirla sin utilizar la palabra clave SmartTags:

```
SetValue Tag1, 64
```

o

```
SetValue SmartTags("Tag1"), 64
```

- Referencias a otros objetos (p.ej. imágenes de proceso, conexiones y ficheros)

Si desea transferir como parámetro una referencia a un objeto, introdúzcala entre comillas:

```
ActivateScreen "MainScreen", 0
```

Particularidades al llamar a scripts

Al llamar a un script, los parámetros se transfieren conforme al principio "Call by Reference". Si transfieren como parámetro p. ej., una variable, las asignaciones de valores en el script tendrán efecto inmediato sobre el valor de la variable.

Dependencia del panel de operador en el script

El código de un script depende del panel de operador seleccionado. Si en el script se utilizan funciones del sistema que no soporten el panel de operador, aparecerá un aviso de error en la ventana de resultados.

4.5 Acceder a objetos

Introducción

Los objetos del modelo de objetos runtime están disponibles en el script junto con las propiedades y los métodos correspondientes.

En runtime es posible leer y modificar las propiedades de los objetos.

Crear referencias a objetos

Para crear referencias a objetos en los scripts, utilice la lista correspondiente. Para identificar un objeto, utilice su nombre o su número de posición en la lista.

La siguiente instrucción permite crear una referencia al primer objeto de la imagen "MainScreen":

```
'VBS_Example_01
Dim objObject
'Change to Screen "MainScreen"
HMIRuntime.BaseScreenName = "MainScreen"
Set objObject = HMIRuntime.Screens(1).ScreenItems(1)
```

La siguiente instrucción sirve para crear una referencia con un objeto mediante su nombre, así como para modificar una propiedad del objeto. Para ello deberá haber creado el objeto con este nombre en la imagen.

```
'VBS_Example_02
Dim objCircle
HMIRuntime.BaseScreenName = "MainScreen"
Set objCircle = HMIRuntime.Screens(1).ScreenItems("Circle_01")
objCircle.BackColor = vbGreen
```

4.6 Sincronizar variables y objetos

Introducción

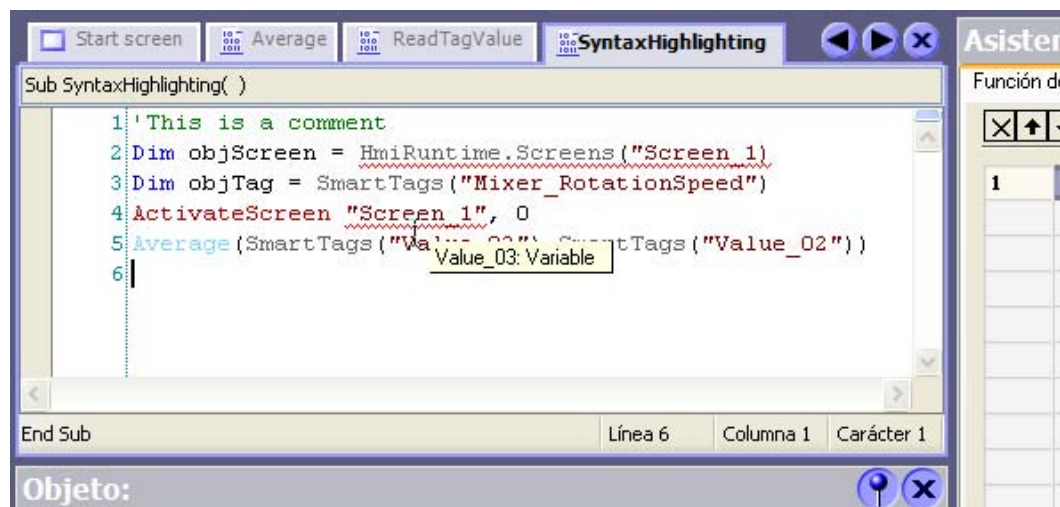
Si modifica el nombre de un objeto en WinCC flexible, dicho cambio tendrá efecto en el proyecto entero. En los scripts, estos cambios se denominan "sincronizaciones".

Ejemplo de aplicación

En el editor "Variables" ha definido la variable "OilTemperature". Esta variable se debe utilizar en un script. Durante la configuración ha cambiado el nombre de la variable a "OilTemperaturMotor1" en el editor "Variables".

- El script estaba abierto al cambiar el nombre:

El nombre antiguo de la variable se subrayará en el script con una línea ondulada azul. Al mover el puntero del ratón sobre el nombre de la variable, aparecerá una información al respecto: Si hace clic en el botón "Sincronizar", el nombre de la variable se cambiará en el script:



- El script estaba cerrado al cambiar el nombre
 Cuando abra el script de nuevo, la variable se sincronizará automáticamente.

4.7 Comprobar scripts

Introducción

El código se comprueba en segundo plano durante la programación. Los errores de sintaxis se identifican con una línea ondulada. En este proceso se comprueba si es correcta la sintaxis y si son válidas las referencias a objetos.

Utilice la función de test del editor de scripts para detectar todos los errores en el código, así como para visualizar los correspondientes avisos de error. Durante el test se emiten también los avisos de error que ha generado el analizador sintáctico de VBScript.

Para comprobar si el script contiene errores de programación lógicos, utilice un depurador por separado.

Tipos de errores

Los tipos de errores indicados a continuación se detectan al comprobar el script y se emiten en la ventana de resultados:

- General:
 - Secuencia de caracteres desconocida (p.ej. no es una palabra clave)
 - Asignación de un valor a una constante
 - Ya existe el nombre de una variable local creada con la instrucción "Dim", p.ej. como parámetro, objeto o elemento del modelo de objetos.
- Modelo de objetos:
 - Muy pocos o demasiados parámetros al utilizar una lista
 - La denominación del objeto no se ha encontrado en la base de datos del proyecto
 - La referencia al objeto no se ha asignado con "Set" de una variable o propiedad
 - El primer denominador de objeto no forma parte del modelo de objetos runtime
 - Utilización de paréntesis en una propiedad sin método "Item"
- Funciones del sistema:
 - Utilización de la notación de objetos en parámetros del tipo "Constant" o "Selection".
 - Tipo incorrecto al transferir el parámetro, p.ej. objeto incorrecto o tipo de datos incorrecto
 - El denominador del objeto transferido no se ha encontrado en la base de datos del proyecto
 - Muy pocos o demasiados parámetros

Requisito

El script se deberá haber creado.

Procedimiento

1. En la ventana del proyecto, haga clic en el script que desea comprobar.
El script se abrirá.
2. En la barra de herramientas, haga clic en el botón "Verificar sintaxis".
Si la sintaxis del script es correcta y si todas las referencias a objetos son válidas, se emitirá un mensaje exitoso en la ventana de resultados.
Si el script contiene errores, todos éstos se indicarán en la ventana de resultados. En el script, el cursor aparecerá en la posición en la que se ha detectado el error.

Ver también

Depurar scripts (Página 5-1)

4.8 Guardar scripts

Principio

Para guardar un script, deberá almacenar el proyecto. Por ejemplo, si desea utilizar un script en un ordenador diferente, podrá copiar el código del script en un archivo de texto, o bien insertarlo en otro proyecto utilizando el método de copiar y pegar.

4.9 Cambiar el nombre de un script

Requisito

El script se deberá haber creado.

Procedimiento

1. En la ventana del proyecto, seleccione con el ratón el script cuyo nombre desea cambiar.
2. En el menú contextual del script, elija el comando "Cambiar nombre".
3. Introduzca un nuevo nombre para el script y pulse la tecla <INTRO>.

Resultado

El nombre del script se cambiará.

Procedimiento alternativo

El nombre del script también se puede cambiar en la ventana de propiedades.

Depurar

5.1 Depurar scripts

Introducción

La depuración permite comprobar en runtime los errores lógicos de programación en los scripts. Así puede comprobar p.ej. si se han transferido los valores correctos a las variables y si las condiciones de interrupción se han efectuado correctamente.

Para depurar los scripts, utilice únicamente el "Microsoft Script Debugger", o bien el "Microsoft Script Editor" suministrado con Microsoft Office XP.

Tipos de errores

Al depurar se detectan los tipos de errores siguientes:

- Errores runtime

Un error runtime se presenta cuando se intenta ejecutar una instrucción no válida o errónea (p.ej. si una variable no está definida).

Para detectar los errores runtime, utilice en el script la instrucción "On Error Resume Next". Tras presentarse un error runtime, esta instrucción hace que se ejecute la instrucción siguiente. En la línea siguiente podrá comprobar el código de error con el objeto "Err". Para volver a desactivar el tratamiento de errores runtime en el script, utilice la instrucción "On Error Goto 0". Para más información sobre el tratamiento de errores, consulte la ayuda en pantalla de Microsoft VBS en el sistema de información.

- Errores lógicos

Un error lógico se presenta cuando no ocurre el evento esperado, p.ej. porque una condición se ha comprobado erróneamente. Para depurar los errores lógicos, verifique todos los pasos del script hasta encontrar la sección defectuosa del mismo.

Ver también

Comprobar scripts (Página 4-8)

5.2 Integrar un depurador

Instalar un depurador de scripts para WinCC flexible

Para buscar errores en los scripts con WinCC flexible es preciso instalar un depurador de scripts.

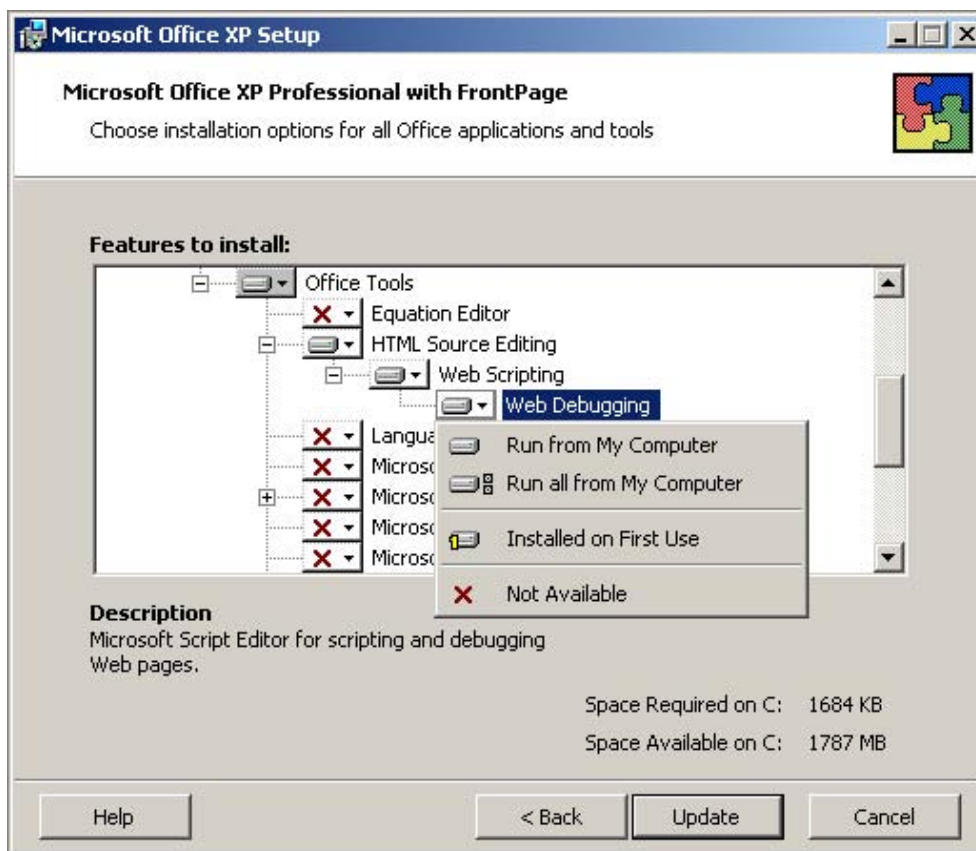
Hemos comprobado y autorizado el uso los siguientes depuradores de scripts:

- Microsoft Script Editor de Office XP
- Microsoft Script Debugger

Si se presenta un error runtime, el depurador de scripts instalado arrancará automáticamente al ejecutarse el script, o bien con el comando "Start runtime system with script debugger".

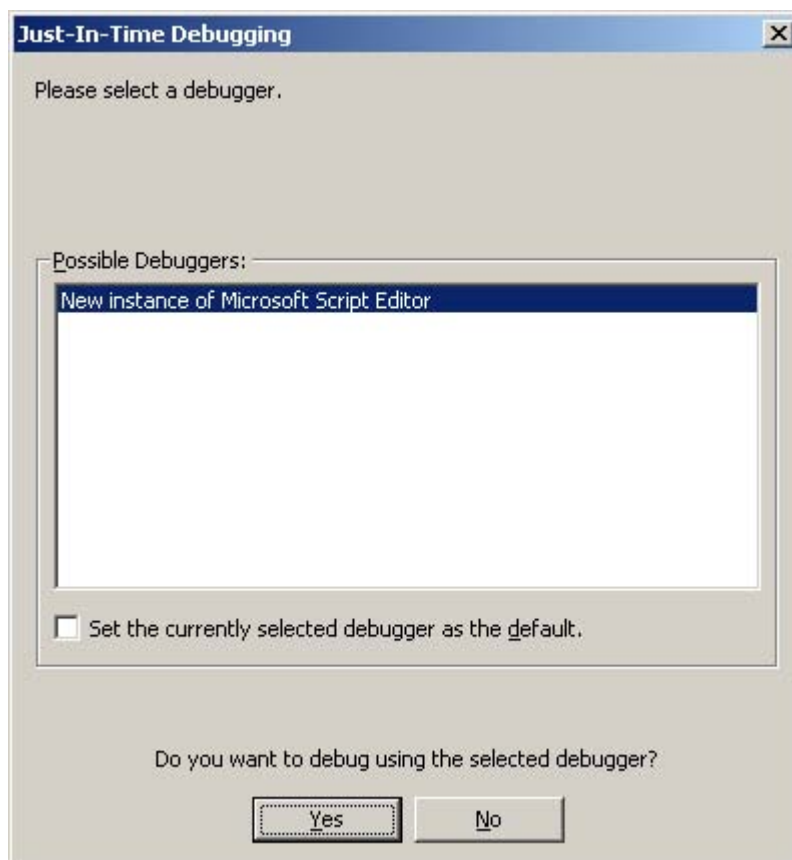
Microsoft Script Editor

El "Microsoft Script Editor" que se suministra junto con Microsoft Office XP incorpora un depurador de scripts. Si Microsoft Office se ha instalado con la configuración estándar, el componente "Microsoft Script Editor" estará configurado de manera que se instale apenas cuando se necesite por primera vez ("Installed on First Use"). Si desea instalar explícitamente este componente, deberá indicarlo durante el proceso de instalación de Microsoft Office. En el menú contextual del componente, haga clic en "Web Debugging" y seleccione la opción "Run from My Computer".

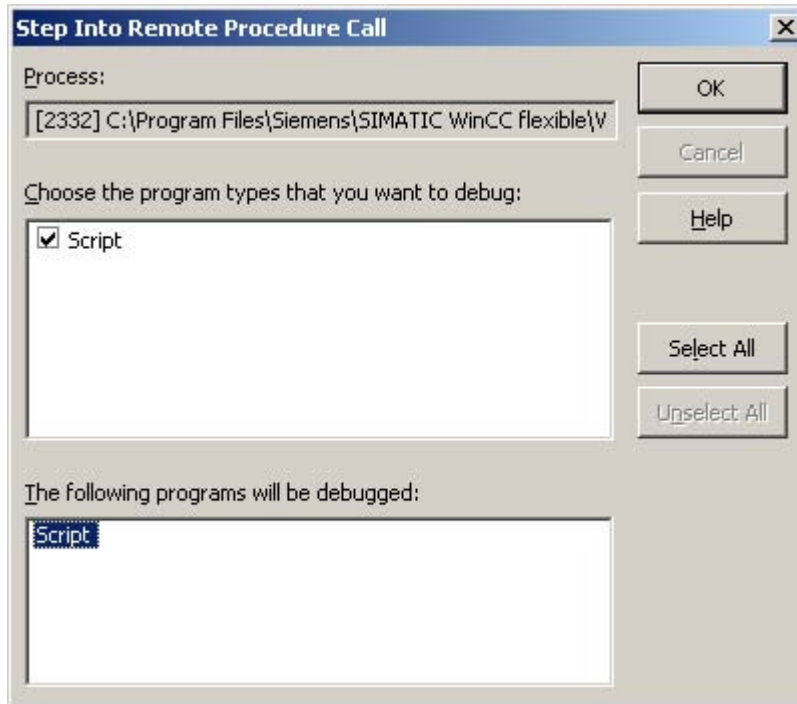


Si un proyecto se inicia en WinCC flexible con el comando "Start runtime system with script debugger", al ejecutar el primer script aparecerá un cuadro de diálogo con una lista de los depuradores de scripts disponibles.

En dicha lista pueden figurar también otros depuradores de scripts instalados, tales como "Microsoft Visual Interdev" o "Microsoft Visual Studio .NET". Seleccione el "Microsoft Script Editor" y confirme su selección con "Yes".



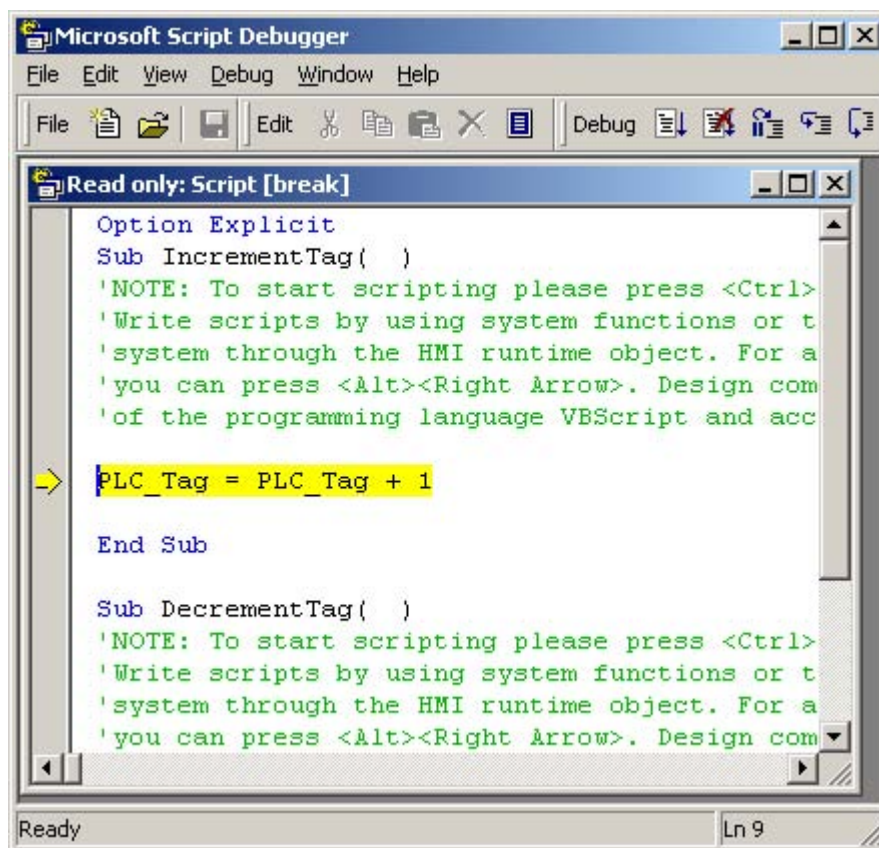
En el cuadro de diálogo "Step Into Remote Procedure Call", active el objeto "Script" y confirme haciendo clic en "OK".



El "Microsoft Script Editor" se iniciará y la ejecución se detendrá en la primera línea del primer script.

Microsoft Script Debugger

Si no dispone de un depurador de scripts, podrá descargar gratuitamente el "Microsoft Script Debugger" (scd10en.exe) de la página web de Microsoft (www.microsoft.com). WinCC flexible iniciará automáticamente el depurador una vez instalado éste último.



Nota

El "Microsoft Script Debugger" no se soportará si en el ordenador ya está instalado otro depurador de scripts.

El depurador de scripts no arranca al inicio de runtime

Si ha instalado un depurador de scripts y, sin embargo, éste no arranca al "Iniciar runtime con el depurador de scripts", compruebe los ajustes siguientes en el Registro de Windows:

1. El ajuste siguiente deberá estar activado para poder realizar el "Just in Time Debugging":
[HKEY_CURRENT_USER\Software\Microsoft\Windows Script\Settings]
"JITDebug"=dword:00000001
Si modifica este ajuste, los cambios tendrán efecto sólo tras reiniciar el equipo.
2. Para inhibir la depuración del Internet Explorer deberá activar el ajuste siguiente:
[HKEY_CURRENT_USER\Software\Microsoft\Internet Explorer\Main]
"Disable Script Debugger" = "yes"

5.3 Iniciar el depurador

Requisito

Un depurador apto para VBS (p.ej. el Microsoft Script Debugger) y WinCC flexible Runtime deberán estar instalados en el PC de configuración.

El proyecto deberá estar abierto.

Procedimiento

1. En la barra de herramientas, haga clic en el botón "Iniciar WinCC flexible con el depurador de scripts".

El software runtime buscará los depuradores instalados en el PC de configuración.

2. Si se han encontrado varios, haga clic en el depurador deseado.

Resultado

El depurador se vinculará automáticamente al software runtime.

Procedimientos alternativos

El depurador también se puede iniciar en el cuadro de diálogo "Ejecutar" de MS Windows. Introduzca para ello el comando siguiente: "HmiRtm /ScriptDebug /<archivo de configuración>".

En el Explorador de Windows puede elegir el comando "Depurar" en el menú contextual del archivo de configuración.

5.4 Manejar el depurador

Procedimiento

El manejo se describe en la documentación del depurador utilizado.

5.5 Salir del depurador

Procedimiento

El depurador no se cierra automáticamente al salir del software runtime. Por tanto, deberá cerrarlo por separado.

Comportamiento al ejecutar funciones en runtime

6.1 Procesar la lista de funciones en runtime

Principio

Las listas de funciones se procesan en runtime de forma secuencial, es decir, de arriba a abajo. Para evitar tiempos de espera durante el procesamiento, se distingue entre los tipos de procesamiento síncrono y asíncrono. El sistema efectúa esta distinción, evaluando para ello los diversos tiempos de ejecución de las funciones del sistema. Independientemente del tiempo de ejecución, los scripts se procesan siempre de forma síncrona. El procesamiento de la lista de funciones se interrumpirá si una función del sistema retorna un estado de error.

Procesamiento síncrono

En el procesamiento síncrono, las funciones del sistema contenidas en una lista de funciones se procesan de forma sucesiva. Para que una función del sistema se pueda ejecutar, la función anterior deberá haber finalizado.

Procesamiento asíncrono

El tiempo de ejecución de las funciones del sistema que efectúan operaciones con archivos (tales como guardar o leer) es más prolongado que el de las funciones que p.ej. activan un valor de variable.

Por tanto, las funciones del sistema de ejecución prolongada se procesan de forma asíncrona. Por ejemplo, mientras que una función del sistema escribe un registro de receta en un soporte de datos, se ejecuta ya la siguiente función. Gracias al procesamiento paralelo de las funciones del sistema se evitan tiempos de espera en el panel de operador.

6.2 Procesar scripts en runtime

Principio

En runtime no se puede procesar más de un script al mismo tiempo. Si se deben procesar varios scripts, éstos se pondrán en una cola de espera y se procesarán uno tras otro.

Nota

Un bucle en un script bloquea de este modo la ejecución de los restantes scripts en cola de espera, aunque los scripts hayan sido iniciados de forma asíncrona.

WinCC flexible soporta una profundidad de anidamiento de 8 scripts como máximo. Tenga en cuenta que la profundidad de anidamiento no se comprueba.

Nota

Si asocia un script al evento "Parada de runtime", sólo podrá utilizar en dicho script aquellas funciones del sistema que se hayan definido como objeto configurable en la referencia "Parada de runtime".

Vigile que la parada de runtime no se vea afectada por la ejecución del script.

6.3 Transferir y retornar valores

Transferir un valor

Al llamar a un script, los parámetros se transfieren conforme al principio "Call by Reference". Si transfiere como parámetro p. ej., una variable, las asignaciones de valores en el script tendrán efecto inmediato sobre el valor de la variable.

En el script no es necesario crear variables locales para los parámetros, siendo posible utilizar allí los parámetros directamente.

Ejemplo: La función del sistema "FijarValor(Y, X)" le asigna el valor "5" a la variable "IndexTag":

```
FijarValor IndexTag, 5
```

Retornar un valor

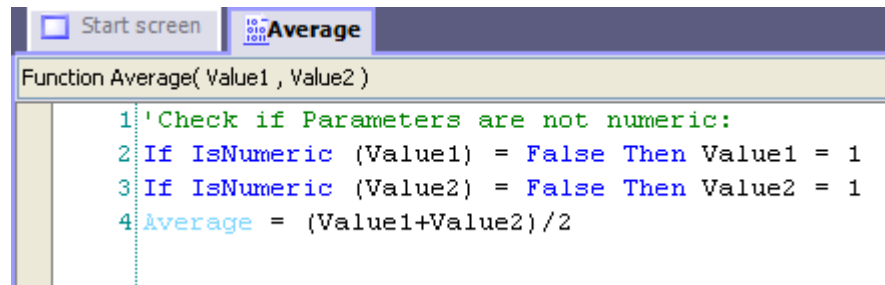
Los valores de retorno sirven para retornar el resultado de un cálculo (p.ej. el valor medio de dos números). Asimismo permiten indicar si una instrucción se ha ejecutado correctamente.

Por este motivo, las funciones del sistema que ejecutan operaciones con archivos (p.ej. borrar) tienen también un valor de retorno.

Nota

El valor de retorno de una función del sistema sólo se puede asignar a una variable externa o interna.

Para que un script pueda retornar un valor, deberá haber seleccionado el tipo "Function". En el script, asigne el valor de retorno al nombre del script:



```
Function Average( Value1 , Value2 )  
1 'Check if Parameters are not numeric:  
2 If IsNumeric (Value1) = False Then Value1 = 1  
3 If IsNumeric (Value2) = False Then Value2 = 1  
4 Average = (Value1+Value2)/2
```

Para calcular el valor medio de dos números, llame a la función "Average" y transfiera p.ej. a una variable los valores que desea procesar:

```
AverageValue = Average (4, 6)
```

A continuación podrá leer el valor medio en un campo de salida.

6.4 Modificar propiedades de objetos en runtime con VBS

Introducción

Con VBS se puede acceder en runtime a propiedades de objetos de imagen y a variables. Si modifica los valores de las propiedades de objetos con VBS, ello no tendrá efecto en los datos del proyecto.

Modificar las propiedades de objetos

Si modifica en runtime una propiedad de un elemento de imagen con VBS, este cambio sólo será efectivo mientras que la imagen esté activa. Cuando cambie de imagen o cargue la imagen de nuevo, se visualizarán de nuevo las propiedades configuradas.

Cambio de idioma

Si cambia el idioma en runtime, los textos de los datos de configuración se cargarán en el nuevo idioma. Si ha modificado textos con VBS, éstos se sobrescribirán.

6.5 Funciones del sistema dependientes del panel de operador

Principio

Si utiliza en un script funciones del sistema no disponibles en el panel de operador seleccionado, aparecerá un aviso de advertencia. Además, la función del sistema en cuestión se subrayará en el script con una línea ondulada en color azul.

Ejemplos

7.1 Ejemplo: Función para convertir temperaturas de Fahrenheit a Celsius

Objetivo

En el presente ejemplo se crea un script que convierte el valor de una sonda de temperatura de Fahrenheit a Celsius (grados centígrados). La temperatura se indica en el panel de operador en un campo de salida.

Configuración

Para este ejemplo se necesitan dos variables y un script configurados como se indica a continuación:

Variables:

Nombre	Conexión al autómat	Tipo
Fahrenheit	sí	Real
Celsius	no	Float

Script:

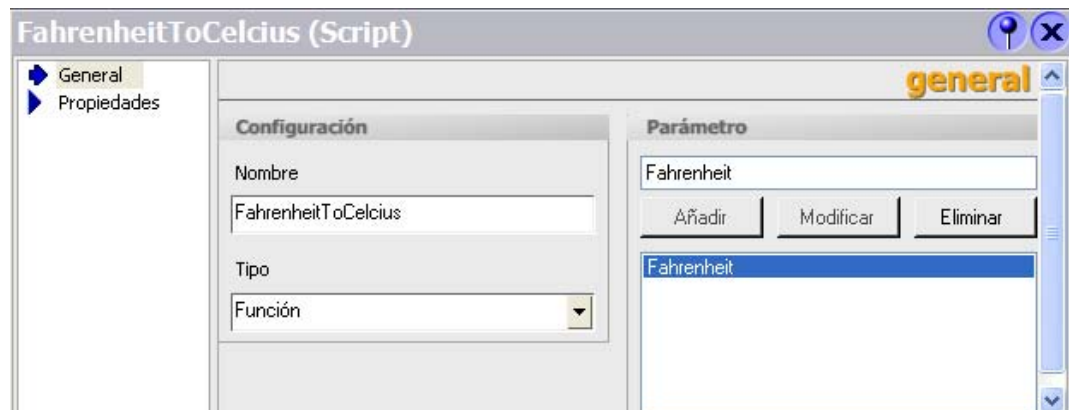
Nombre	Tipo	Parámetro
FahrenheitToCelsius	Function	Fahrenheit

Procedimiento

1. Cree las dos variables "Fahrenheit" y "Celsius" con la configuración indicada.



2. Cree el script "FahrenheitToCelsius" con la configuración indicada.



3. Introduzca el código siguiente en el script:

FahrenheitToCelsius = (Fahrenheit-32) * (5/9)

```
Function FahrenheitToCelsius( Fahrenheit )  
1 FahrenheitToCelsius = (Fahrenheit-32) * (5/9)
```

Resultado intermedio

Las variables y el script se crearán.

Procedimiento

1. Abra el editor "Variables" y haga clic en la variable "Fahrenheit".
2. En la ventana de propiedades del grupo "Eventos", haga clic en "Modificar".
3. Asocie el script "FahrenheitToCelsius" al evento "Modificar".
4. Seleccione la variable "Fahrenheit" para el parámetro "Fahrenheit".
5. Seleccione la variable "Celsius" para el valor de retorno "FahrenheitToCelsius".
6. Configure un campo de salida en una imagen de proceso y asícielo a la variable "Celsius".

Procedimiento alternativo

En lugar de utilizar un script del tipo "Function", se puede utilizar el tipo "Procedure". En este caso, asigne directamente el valor convertido a la variable "Celsius":

SmartTags("Celsius") = (Fahrenheit-32) * (5/9) Con este procedimiento, se suprime el valor de respuesta del script.

Resultado

Si el valor de la variable "Fahrenheit" cambia en runtime, se procesará el script "FahrenheitToCelsius". El valor de temperatura convertido se retornará a la variable "Celsius" y se visualizará en el campo de salida.

7.2 Ejemplo: Subrutina para convertir pulgadas a metros

Objetivo

En el presente ejemplo se crea un script que convierte de pulgadas a metros el valor de un medidor de longitud. La longitud en metros se indica en el panel de operador en un campo de salida.

Configuración

Para este ejemplo se necesitan dos variables y un script configurados como se indica a continuación:

Variables:

Nombre	Conexión al autómatas	Tipo
VarInch	sí	Real
VarMeter	no	Float

Script:

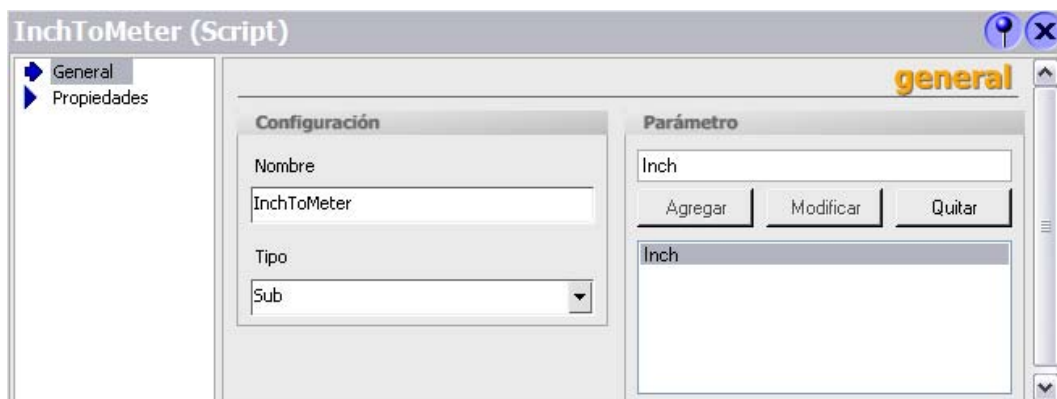
Nombre	Tipo	Parámetro
InchToMeter	Sub	Inch

Procedimiento

1. Cree las dos variables "VarInch" y "VarMeter" con la configuración indicada.

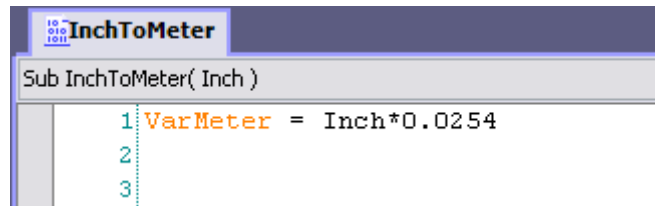


2. Cree el script "InchToMeter" con la configuración indicada.



3. Introduzca el código siguiente en el script:

```
VarMeter = Inch*0.0254
```



Resultado intermedio

Las variables y el script se crearán.

Procedimiento

1. Abra el editor "Variables" y haga clic en la variable "VarInch".
2. En el grupo "Eventos" de la ventana de propiedades, haga clic en "Modificación de valor".
3. Asocie el script "InchToMeter" al evento "Modificación de valor".
4. Seleccione la variable "Inch" para el parámetro "VarInch".
5. Configure un campo de salida en una imagen de proceso y asícielo a la variable "VarMeter".

Resultado

Si el valor de la variable "VarInch" cambia en runtime, se procesará el script "InchToMeter". El valor convertido se escribirá en la variable "VarMeter" y se visualizará en el campo de salida.

7.3 Ejemplo: Cambiar el modo de operación del panel de operador y visualizar el estado actual

Objetivo

En el presente ejemplo se utiliza la función del sistema "EstablecerModoDeOperacionEquipo" para conmutar entre los modos de operación "online" y "offline" en el panel de operador. Además, el modo de operación ajustado actualmente se debe visualizar en el panel de operador.

Requisito

El panel de operador deberá disponer de teclas de función. Se deberá haber creado una imagen de proceso.

Configuración

Para este ejemplo se necesitan una variable y una lista de texto configuradas como se indica a continuación:

Variable:

Nombre	Conexión al autómeta	Tipo
OperatingMode	no	Bool

Lista de texto:

Nombre	Contiene	Valores
ShowOperatingMode	Boolean (0/1)	1: Modo de operación: "Online": 0: Modo de operación: "Offline"

Procedimiento

1. Cree la variable "OperatingMode" como se indica arriba.

The screenshot shows the 'VARIABLES' configuration window with the following data:

Nombre	Conexión	Tipo de datos	Dirección	Elementos de matriz	Ciclo de adq
Value_02	<Variable interna>	Int	<Ninguna dirección>	1	1 s
Value_00	<Variable interna>	Int	<Ninguna dirección>	1	1 s
OperatingMode	<Variable interna>	Bool	<Ninguna dirección>	1	1 s
Mixer_Rotation_1	SIMATIC S7 300_400	Int	DB 1 DBW 0	1	1 s

2. Cree la lista de texto "ShowOperatingMode" como se indica arriba.

The screenshot shows the 'LISTAS DE TEXTO' configuration window with the following data:

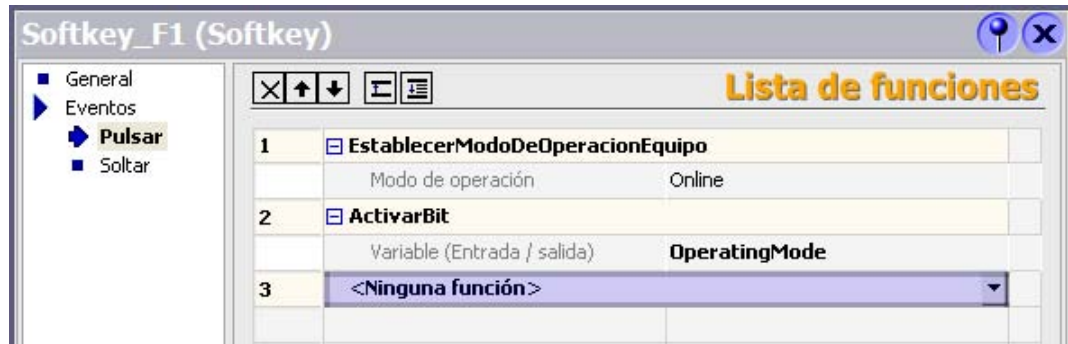
Nombre	Selección	Comentario
ShowOperatingMode	Bit (0, 1)	

Valor	Entrada
0	Modo de operación: Offline
1	Modo de operación: Online

25 / 320

3. Abra la imagen de proceso y haga clic en la tecla de función en la que desea configurar el cambio de modo de operación a "online".
4. En la ventana de propiedades del grupo "Eventos", haga clic en "Pulsar".

5. Configure la función del sistema "EstablecerModoDeOperacionEquipo" asociada al evento "Pulsar". Esta función del sistema se encuentra en la lista de selección bajo "Configuración".
6. En la lista, seleccione la entrada "Online" para el parámetro "Modo".
7. Configure la función del sistema "ActivarBit" asociada al evento "Pulsar". Esta función del sistema se encuentra en la lista de selección bajo "Procesamiento de bits".
8. En la lista, seleccione la variable "OperatingMode" para el parámetro "Variable".



9. En la imagen de proceso, haga clic en la tecla de función en la que desea configurar el cambio de modo de operación a "offline".
10. Repita los pasos 4 a 7. En la lista, seleccione la entrada "Offline" para el parámetro "Modo". Configure la función del sistema "RestablecerBit" en vez de la función "ActivarBit".



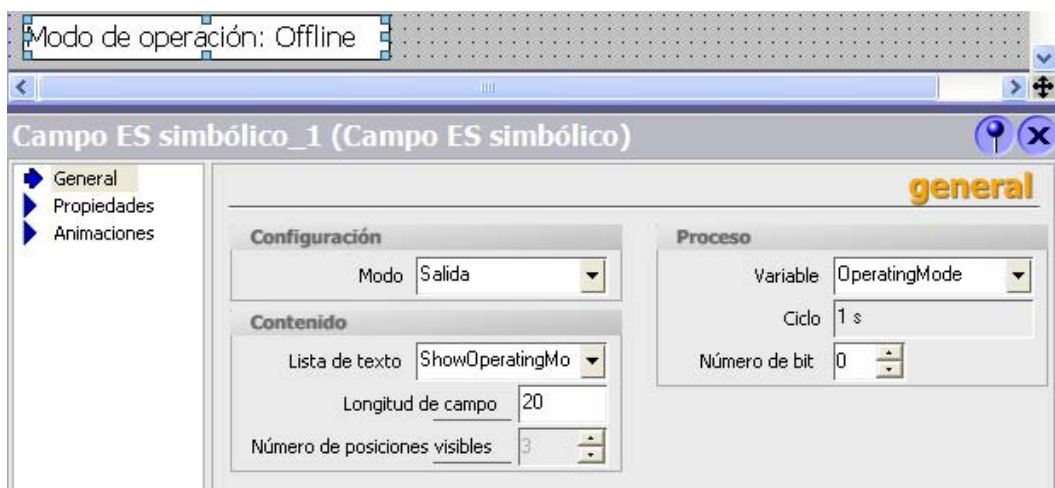
Resultado intermedio

El modo de operación del panel de operador se podrá cambiar en runtime mediante ambas teclas de función.

El modo de operación ajustado actualmente se debe visualizar en un campo de salida en el panel de operador.

Procedimiento

1. En la imagen de proceso, cree un "Campo ES simbólico" y, en la ventana de propiedades, efectúe los ajustes siguientes en el grupo "General":
 - Bajo "Modo", seleccione "Salida".
 - Seleccione la lista de texto "ShowOperatingMode".
 - Seleccione la variable "OperatingMode".



Resultado

Si cambia el modo de operación utilizando las teclas de función, el estado ajustado actualmente se visualizará en el panel de operador.

Índice alfabético

A

- Abrir
 - Editor de scripts, 3-1
- Acceder al modelo de objetos runtime, 4-6
- Acceso
 - Variable, 4-4
- Área de trabajo
 - Editor "Scripts", 3-2
- Arrastrar y soltar, 3-5
- Asistente de scripts, 3-2
 - configurar, 2-3

C

- cambiar nombre
 - Script, 4-9
- Cambiar y visualizar el modo de operación (ejemplo), 7-4
- Cambio de idioma
 - En runtime, 6-3
- Característica
 - Editor de scripts, 3-3
 - Lista de funciones, 2-2
- comprobar
 - Script, 4-8
- Configuración básica
 - Script, 3-6
- configurar
 - Asistente de scripts, 2-3
 - Lista de funciones, 2-3
 - Script, 4-1
- Crear
 - Script, 4-1
- crear referencias
 - Objeto, 4-6

D

- Dependencia del idioma
 - Función del sistema, 4-5
- Dependencia del panel de operador
 - en un script, 4-6

- Depurador, 5-1
 - Iniciar, 5-6
 - salir, 5-6
 - Tipos de errores, 5-1
- Depurar, 5-1

E

- Edición
 - Lista de funciones, 2-5
- Editor
 - Script, 3-1
- Editor de scripts
 - Modificar el resalto de sintaxis, 3-7
- Editor de scripts, 3-1
 - Abrir, 3-1
 - Área de trabajo, 3-2
 - Características, 3-3
 - Modificar la configuración, 3-6
- Ejecutar
 - Función del sistema en un script, 4-2, 4-5
 - Script en script, 4-2, 4-5
- Ejemplo
 - Cambiar y visualizar el modo de operación, 7-4
 - Crear una subrutina, 7-3
 - Introducir una función para convertir la temperatura, 7-1
- error lógico, 5-1
- Errores
 - error lógico, 5-1
 - Errores runtime, 5-1
- Errores runtime, 5-1

F

- Función de ayuda, 3-5
- Función del sistema, 1-1, 1-2
 - Dependencia del idioma, 1-3, 4-5
 - En listas de funciones, 1-4, 2-2
 - en scripts, 1-4
 - en un script, 4-5, 6-3
 - llamar en el script, 4-2, 4-5
 - Particularidades al llamar, 4-5
 - Posibilidades de aplicación, 1-4

Transferir parámetros, 4-5
Utilización, 1-3, 1-4

I

Información de estado
 Lista de funciones, 2-2
Iniciar
 Depurador, 5-6
IntelliSense, 3-3
Introducir una función para convertir la temperatura (ejemplo), 7-1

L

Lista de funciones, 2-1
 Característica, 2-2
 configurar, 2-3
 Dependencia del panel de operador, 2-2
 editar, 2-5
 Función del sistema, 2-2
 Información de estado, 2-2
 Procesamiento, 2-2
 Procesamiento asíncrono, 6-1
 Procesamiento en runtime, 6-1
 Procesamiento síncrono, 6-1
 Script, 2-2

M

Marcación de sintaxis, 3-4
 Modificar, 3-7
Modificar
 Formato de código en el script, 3-7
 Marcación de sintaxis, 3-7
 Propiedades de objetos, 6-3

O

Objeto
 Acceso, 4-6
 crear referencias, 4-6
 Modificar propiedades con VBS, 6-3
 sincronizar en un script, 4-7
Organizar
 Script, 1-5

P

Procesar
 Scripts en runtime, 6-2

Profundidad de recurrencia, 1-2

R

Runtime
 Cambio de idioma, 6-3
 Modificar propiedades de objetos con VBS, 6-3
 Procesar la lista de funciones, 6-1
 Procesar scripts, 6-2
 Transferir parámetros, 6-2
Runtime Scripting, 1-1
 Aplicación, 1-1

S

salir
 Depurador, 5-6
Script, 1-2, 1-4
 Cambiar nombre, 4-9
 Características, 1-5
 comprobar, 4-8
 Configuración básica, 3-6
 configurar, 4-1
 Crear, 4-1
 Dependencia del panel de operador, 4-6
 Depurar, 5-1
 En listas de funciones, 2-2
 en un script, 4-5
 Función de ayuda, 3-5
 Introducir el código, 4-2
 llamar en el script, 4-2, 4-5
 Modificar el formato de código, 3-7
 Organizar, 1-5
 Particularidades al llamar, 4-6
 Procesar en runtime, 6-2
 Profundidad de recurrencia, 1-2
 Transferir parámetros, 4-6
 Utilización, 1-6
 Utilizar funciones del sistema, 6-3
 Utilizar variables, 4-2
 Valor de retorno, 6-2
Sincronización automática, 4-7
Sincronización manual, 4-7
Sincronizar
 automática, 4-7
 Manual, 4-7
 manualmente, 4-7
 objetos en un script, 4-7
 variables en un script, 4-7
Subrutina
 Crear, 7-3

T

- Transferencia
 - Eliminar comentarios del script, 3-7
 - No transferir nombres de objeto, 3-7
- Transferir parámetros
 - Función del sistema, 4-5
 - Script, 4-6
- Transferir parámetros en runtime, 6-2

U

- Utilización
 - Función del sistema, 1-4
 - Script, 1-6

V

- Valor de retorno, 6-2
- Variable
 - Acceder con VBS, 4-4
 - local, 4-4
 - sincronizar en un script, 4-7
 - utilizar en scripts, 4-2
- Variable del proyecto, 4-4
- Variable externa, 4-4
- Variable interna, 4-4
- variable local, 4-4
- VBS
 - Función de ayuda, 3-5
 - Modificar propiedades de objetos, 6-3

